B2Z10030
Online Ordering System
Database Best Practices
Attachment #6

- When designing a database keep 'performance' in mind. Make sure queries do 'Index seeks' instead of 'Index scans' or 'Table scans'.

- Normalize to at least to 3rd normal form. If there is a deliberate reason to denormalize such as a need to improve performance, the decision to denormalize must be thoroughly documented.

- Perform all referential integrity checks, data validations using constraints (foreign key and check constraints).

- Follow a database naming convention. Names should be meaningful. Provide the naming convention.

- Data entry, updates and deletions should be efficient.

- Data retrieval, summarization and reporting should be efficient.

- Do not depend on undocumented functionality.

- Do not use system tables directly.

- Use Unicode datatypes like nchar, nvarchar, ntext, only when they are absolutely needed.

- Write comments in stored procedures, triggers and SQL batches generously, whenever something is not very obvious.

- Avoid the use of SELECT * in queries. Always write the required column names after a SELECT statement.

- Avoid the use of cursors as much as possible.

- Avoid the creation of temporary tables while processing data

- Avoid the use of wildcard characters at the beginning of a word while searching using the LIKE keyword.

- Use SET NOCOUNT ON at the beginning of SQL batches, stored procedures and triggers.

- Use ANSI-SQL 99 join clauses in queries.

- Do not prefix stored procedure names with 'sp_'.

- Front-end applications should not query/manipulate the data directly using SELECT or INSERT/UPDATE/DELETE statements.

- Binary files, image files (Binary large objects or BLOBs) etc. should not be stored inside the database. Instead store the path to the binary/image file in the database and use that as a pointer to the actual binary file.

- Avoid the use of char data type for a column when the column is nullable.

- Avoid dynamic SQL statements as much as possible.

- Minimize the usage of NULLs.

- Always use a column list in INSERT statements.

- Attempt to access tables in the same order in all stored procedures/triggers consistently.

- Keep transactions as short as possible.

- Avoid the use of higher level locking hints or restrictive isolation.

- Design front-end applications to be deadlock-intelligent. These applications should be able to resubmit transactions in case the previous transaction fails with error 1205.

- Avoid the performance of tasks such as string manipulations, concatenations, row numbering, case conversions, type conversions etc. inside the database.

- Perform basic data validations prior to accessing the database.

- Always store 4 digit years in dates.